Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1 – 28 (canceled)

Claim 29 (previously presented): A hardware system comprising:

a first processor and a second processor each having a register file, a scoreboard and a decoder;

a bus coupled to the first processor and the second processor;

a main memory coupled to the bus;

a first buffer coupled to the first processor and the second processor to transfer register values from the first processor to the second processor;

a second buffer coupled to the first processor and the second processor to transfer values from the second processor to the first processor; and

wherein the second processor is to execute a portion of instructions of an application ahead of execution of a current instruction of the application by the first processor, and the first processor is to fetch, issue, and avoid execution of the portion of instructions by commitment of results of the portion of instructions into the register file of the first processor from the second buffer.

Claim 30 (canceled)

Claim 31 (previously presented): The system of claim 29, wherein the first processor is coupled to at least one of a plurality of zero level (L0) data caches and at least one of a plurality of L0 instruction caches, and the second processor is coupled to at least one of the plurality of L0 data caches and at least one of the plurality of L0 instruction caches.

Claim 32 (previously presented): The system of claim 31, wherein each of the plurality of L0 data caches is to store exact copies of store instruction data.

Claim 33 (previously presented): The system of claim 31, wherein the first processor and the second processor each share a first level (L1) cache and a second level (L2) cache.

Claim 34 (previously presented): The system of claim 29, further comprising a plurality of memory instruction buffers including at least one store forwarding buffer and at least one load ordering buffer.

Claim 35 (previously presented): The system of claim 34, wherein the at least one store forwarding buffer includes a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

Claim 36 (previously presented): The system of claim 29, wherein the first processor is to commit results in one commit cycle based at least on information received from the second processor.

Claim 37 (currently amended): [[An]] A hardware apparatus comprising:

a first processor and a second processor each having a scoreboard and a decoder;

a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer that is operable to transfer register values from the first processor to the second processor;

a second buffer coupled to the first processor and the second processor; and

wherein the first processor is to direct the second processor to execute a portion of instructions of a single threaded application ahead of a current instruction, of the single threaded application, executed by the first processor, wherein the first processor is to fetch, issue, and avoid execution of the portion of instructions by commitment of results of the portion of instructions into a register file from the second buffer.

Claim 38 (previously presented): The apparatus of claim 37, further comprising a plurality of caches coupled to the first and second processors.

Claim 39 (previously presented): The apparatus of claim 37, wherein the first processor is coupled to at least one of a plurality of zero level (L0) data caches and at least one of a plurality of L0 instruction caches, and the second processor is coupled to at least one of the plurality of L0 data caches and at least one of the plurality of L0 instruction caches.

Claim 40 (previously presented): The apparatus of claim 39, wherein each of the plurality of L0 data caches is to store exact copies of store instruction data.

Claim 41 (previously presented): The apparatus of claim 37, further comprising a plurality of memory instruction buffers including at least one store forwarding buffer and at least one load ordering buffer.

Claim 42 (previously presented): The apparatus of claim 41, wherein the at least one store forwarding buffer comprises a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

Claim 43 (previously presented): The apparatus of claim 42, wherein the at least one load ordering buffer comprises a structure having a plurality of entries, each of the plurality of entries having a tag portion, an entry validity portion, a load identification (ID) portion, and a load thread ID portion.

Claim 44 (previously presented): The apparatus of claim 37, wherein the register buffer comprises an integer register buffer and a predicate register buffer.

Claim 45 (previously presented): A method comprising:

directing, by a first processor, a second processor to execute a plurality of instructions in a thread, wherein the plurality of instructions is at a location in a stream of the thread ahead of a current instruction executed in the first processor;

receiving control flow information from the second processor in the first processor to avoid branch prediction in the first processor; and

receiving results from the second processor in the first processor so that the first processor fetches, issues, and avoids execution of the plurality of instructions by committing the results of the plurality of instructions into a register file of the first processor from a first buffer.

Claim 46 (previously presented): The method of claim 45, further comprising tracking at least one register that is one of loaded from a register file buffer and written by said second processor, said tracking executed by said second processor.

Claim 47 (previously presented) The method of claim 45, further comprising clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer.

Claim 48 (previously presented): The method of claim 45, further comprising:

setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion in a load buffer.

Claim 49 (previously presented): The method of claim 45, further comprising:

flushing a pipeline, setting a mispredicted bit in a load entry in the first buffer and restarting a load instruction if the load instruction is not replayed and does not match a tag portion in a load buffer, or the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 50 (previously presented): The method of claim 45, further comprising:

executing a replay mode at a first instruction of a speculative thread.

Claim 51 (previously presented): The method of claim 45, further comprising:

supplying names from the first buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOP) instructions;

issuing all load instructions and store instructions to memory; and

committing non-replayed instructions from the first buffer to the register file.

Claim 52 (previously presented): The method of claim 45, further comprising:

clearing a valid bit in a first entry in a load buffer if an instruction corresponding to the first entry is retired.

Claim 53 (previously presented): An article comprising a machine-readable storage medium containing first instructions which, when executed by a machine, cause the machine to perform operations comprising:

directing, by a first processor, a second processor to execute a plurality of instructions in a thread, wherein the plurality of instructions is at a location in a stream of the thread ahead of a current instruction executed in the first processor;

receiving control flow information from the second processor in the first processor to avoid branch prediction in the first processor; and

receiving results from the second processor in the first processor so that the first processor fetches, issues, and avoids execution of the plurality of instructions by committing the results of the plurality of instructions into a register file of the first processor from a first buffer.

Claim 54 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations comprising tracking at least one register that is one of loaded from a register file buffer and written by said second processor, said tracking executed by said first processor.

Claim 55 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations comprising clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer.

Claim 56 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

Claim 57 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion.

Claim 58 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

flushing a pipeline, setting a mispredicted bit in a load entry in a second buffer and restarting a load instruction if the load instruction is not replayed and does not match a tag portion in a load buffer, or the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 59 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

executing a replay mode at a first instruction of a speculative thread;

terminating the replay mode and the execution of the speculative thread if a partition in the first buffer is approaching an empty state.

Claim 60 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

supplying names from the first buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOP) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the first buffer to the register file.

Claim 61 (previously presented): The article of claim 53, wherein the first instructions further cause the machine to perform operations including:

clearing a valid bit in a first entry in a load buffer if an instruction corresponding to the first entry is retired.